Activity: **7.5**
Conduct Unit Testing

Responsibility: Project Team Programmers

Description: Unit testing is used to verify the input and output for each module. Successful testing indicates the validity of the function or subfunction performed by the module and shows traceability to the design. During unit testing, each module is tested individually and the module interface is verified for consistency with the design specification. All important processing paths through the module are tested for expected results. All error handling paths are also tested.

Unit testing is driven by test cases and test data that are designed to verify software requirements, and to exercise all program functions, edits, in-bound and out-of-bound values, and error conditions identified in the program specifications. If timing is an important characteristic of the module, tests should be generated that measure time critical paths in average and worst-case situations.

Plan and document the inputs and expected outputs for all test cases in advance of the tests. Log all test results. Analyze and correct all errors and retest the unit using the scenarios defined in the test cases. Repeat testing until all errors have been corrected.

While unit testing is generally considered the responsibility of the programmer, the project manager or lead programmer should be aware of the unit test results.

Work Products: Completion of unit testing for a software component signifies internal project delivery of a component or module for integration with other components. Place all components that have completed unit testing under configuration control as described in the Software Configuration Management Plan. These components form the Production Baseline. Configuration controls restrict changes to tested and approved software in the Production Baseline. Subsequent changes or additions to the software that are agreed upon in a Critical Design Review and receive stakeholder concurrence supersede the existing baseline and establish a new Production Baseline.

Review the draft versions of the Integration and System Test Plans developed during the System Design Stage. Update the plans, as needed, to reflect any changes made to the software design. Deliver the final versions of the Integration and System Test Plans to the system owner and user for review and approval. Place a copy of the approved plans in the Project File.

Work Products, continued: Create a Project Test File for all test materials generated throughout the project lifecycle. Place all unit test materials (e.g., inputs, outputs, results and error logs) in the Project Test File. The test cases used for unit testing may become a subset of tests for integration testing.

Review Process: Conduct peer reviews on the test materials to be placed in the Project Test File. Conduct structured walkthroughs on any updated plans, e.g., Integration and System Test Plans.